

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of ~~searching-in-parallel~~ for a plurality of prime number values ~~substantially simultaneously~~, comprising the steps of:

randomly generating a plurality of  $k$  random odd numbers each providing a prime number candidate; and

~~performing at least one a plurality of  $t$  primality tests on each of said plurality of  $k$  randomly generated prime number candidates, each of said the plurality of  $(k \times t)$  primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $(k \times t)$  of the exponentiation units, said exponentiation operations being performed in parallel by said associated exponentiation units substantially simultaneously.~~

2. (Currently Amended) In a prime number generating system as recited in claim 1 wherein said plurality of  $k$  randomly generated numbers are expressed as  $n_{0,0}, n_{1,0} \dots n_{(k-1),0}$ , further comprising the steps of:

determining a plurality of  $y$  additional odd numbers based on each one of the randomly generated numbers  $n_{0,0}, n_{1,0}, \dots n_{(k-1),0}$  to provide  $(k \times y)$  additional prime number candidates  $(n_{0,1}, n_{0,2}, \dots n_{0,y}), (n_{1,1}, n_{1,2}, \dots n_{1,y}), \dots (n_{(k-1),1}, n_{(k-1),2}, \dots n_{(k-1),y})$  thereby yielding a total number of  $(k \times (y+1))$  prime number candidates;

wherein said step of performing includes performing a primality test on each of said total number  $(k \times (y+1))$  of candidates, each of the plurality of  $(k \times (y+1))$  primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $(k \times (y+1))$  of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of  $(k \times (y+1))$  exponentiation units ~~substantially simultaneously~~.

3. (Original) In a prime number generating system as recited in claim 2 wherein each of said plurality of prime number values being searched for has a specified length, and wherein said plurality of  $y$  additional odd numbers defines an interval that is selected relative to said specified length.

4. (Original) In a prime number generating system as recited in claim 2 wherein said step of determining a plurality of  $y$  additional odd numbers based on each one of the randomly generated numbers  $n_{0,0}, n_{1,0}, \dots, n_{(k-1),0}$  includes successively adding two to each of said randomly generated odd numbers  $n_{0,0}, n_{1,0}, \dots, n_{(k-1),0}$  to provide  $(k \times y)$  additional prime number candidates expressed as  $(n_{0,1} = n_{0,0} + 2, n_{0,2} = n_{0,0} + 4, \dots, n_{0,y} = n_{0,0} + (y \times 2)), (n_{1,1} = n_{1,0} + 2, n_{1,2} = n_{1,0} + 4, \dots, n_{1,y} = n_{1,0} + (y \times 2)), \dots, (n_{(k-1),1} = n_{(k-1),0} + 2, n_{(k-1),2} = n_{(k-1),0} + 4, \dots, n_{(k-1),y} = n_{(k-1),0} + (y \times 2))$ .

5. (Currently Amended) In a prime number generating system as recited in claim 1 wherein said ~~step of performing includes performing a plurality of  $t$  primality tests on each one of said plurality of  $k$  randomly generated numbers, each of the plurality of  $(k \times t)$  primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $(k \times t)$  of the exponentiation units, said exponentiation operations being~~ are performed by said associated exponentiation units substantially simultaneously.

6. (Currently Amended) In a prime number generating system as recited in claim 2 wherein said step of performing includes performing a plurality of  $t$  primality tests on each of said  **$(k \times (y+1))$**  prime number candidates, each of the plurality of  **$(k \times (y+1) \times t)$**  primality tests including an associated exponentiation operation executed by an associated one of a plurality of  **$(k \times (y+1) \times t)$**  of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of  **$(k \times (y+1) \times t)$**  exponentiation units substantially simultaneously.

7. (Currently Amended) In a prime number generating system as recited in claim 1 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing ~~at least one~~ said plurality of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of **s** exponentiation units substantially simultaneously.

8. (Original) In a prime number generating system as recited in claim 7 further comprising the steps of:

receiving a specified public exponent **e** associated with a cryptographic application;  
testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent **e**, wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.

9. (Currently Amended) In a prime number generating system as recited in claim 2 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing at least one primality test on each of said sieved number **s** of candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of **s** exponentiation units substantially simultaneously.

10. (Original) In a prime number generating system as recited in claim 1 further comprising the step of:

sieving said prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality test on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

11. (Original) In a prime number generating system as recited in claim 2 further comprising the step of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to

eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number  $r$  of candidates; and

performing a plurality of  $t-1$  additional primality tests on each of said remaining number  $r$  of candidates, each of the plurality of  $(r \times (t-1))$  primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $(r \times (t-1))$  of the exponentiation units, said  $(r \times (t-1))$  exponentiation operations being performed by said plurality of  $(r \times (t-1))$  exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

12. (Original) In a prime number generating system as recited in claim 1 wherein said step of performing at least one primality test includes performing a Fermat type primality test.

13. (Original) In a prime number generating system as recited in claim 1 wherein said step of performing at least one primality test includes performing a Miller-Rabin type primality test.

14. (Original) In a prime number generating system as recited in claim 1 wherein said step of randomly generating a plurality of  $k$  random odd numbers further includes:

defining a length  $L$  for each of the plurality of  $k$  random numbers to be generated; and

generating each of said plurality of  $k$  random odd numbers in an interval between  $2^L$  and  $2^{L-1}$ .

15. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching in parallel for a plurality of prime number values simultaneously, comprising the steps of.

randomly generating at least one random odd number providing a prime number candidate;

determining a plurality of **y** additional odd numbers based on said at least one randomly generated odd number to provide **y** additional prime number candidates, thereby providing a total number of **y+1** candidates;

performing at least one primality test on each of said **y+1** candidates, each of the **y+1** primality tests including an associated exponentiation operation executed by an associated one of **y+1** of the exponentiation units, said **y+1** exponentiation operations being performed in parallel by said associated **y+1** exponentiation units substantially simultaneously.

16. (Original) In a prime number generating system as recited in claim 15 wherein said at least one randomly generated odd number is expressed as  $n_{0,0}$ , and wherein said step of determining a plurality of **y** additional odd numbers based on said randomly generated odd number  $n_{0,0}$  includes successively adding two to said randomly generated odd number  $n_{0,0}$  to provide **y+1** additional prime number candidates expressed as ( $n_{0,1} = n_{0,0} + 2$ ,  $n_{0,2} = n_{0,0} + 4$ , ...  $n_{0,y} = n_{0,0} + (y \cdot 2)$ ).

17. (Original) In a prime number generating system as recited in claim 15 wherein said step of performing includes performing a plurality of **t** primality tests on each of said **(y+1)** prime number candidates, each of the plurality of **((y+1) x t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **((y+1) x t)** of the exponentiation units, said exponentiation operations being performed by said plurality of **((y+1) x t)** exponentiation units substantially simultaneously.

18. (Original) In a prime number generating system as recited in claim 15 further comprising the step of sieving said **y+1** candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved numbers of candidates.

19. (Original) In a prime number generating system as recited in claim 18 further comprising the step of:

receiving a specified public exponent **e** associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent  $e$ , wherein said step of testing the suitability is performed prior to said step of performing at least one primality test

20. (Original) In a prime number generating system as recited in claim 15 further comprising the step of:

sieving said  $y+1$  candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number  $s$  of candidates;

wherein said step of performing includes performing an associated first one of  $t$  primality test on each of said sieved number  $s$  of candidates, each of the plurality of  $s$  first primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $s$  of the exponentiation units, said first exponentiation operations being performed by said plurality of  $s$  exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number  $r$  of candidates; and

performing a plurality of  $t-1$  additional primality tests on each of said remaining number  $r$  of candidates, each of the plurality of  $(r \times (t-1))$  first primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $(r \times (t-1))$  of the exponentiation units, the  $(r \times (t-1))$  exponentiation operations being performed by said plurality of  $(r \times (t-1))$  exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

21. (Original) In a prime number generating system as recited in claim 15 wherein said step of 2 performing at least one primality test includes performing a Fermat type primality test.

22. (Original) In a prime number generating system as recited in claim 15 wherein said step of performing at least one primality test includes performing a Miller-Rabin type primality test.

23. (Original) In a prime number generating system as recited in claim 15 wherein said step of randomly generating at least one random odd number further includes:

defining a length  $L$  for each ~~of~~ the plurality ~~of~~  $k$  random numbers to be generated; and

generating each of said plurality of  $k$  random odd numbers in an interval between  $2^L$  and  $2^{L-1}$ .

24. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching ~~in parallel~~ for a plurality of prime number values ~~simultaneously~~, comprising the steps of:

randomly generating at least one random odd number providing a prime number candidate; and

testing the primality of said candidate by performing a plurality of  $t$  primality tests on said candidate, each of the plurality of the  $t$  primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $t$  of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of  $t$  exponentiation units substantially simultaneously.

25. (Original) In a prime number generating system as recited in claim 24 further including the step of sieving said candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number  $s$  of candidates.

26. (Original) In a prime number generating system as recited in claim 25 further including the step of:

receiving a specified public exponent  $e$  associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate



minus one and said specified public exponent  $e$ , wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.

27. (Original) In a prime number generating system as recited in claim 24 wherein said step of testing the primality of said candidate further includes:

sieving said candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said 5 small divisor test thereby yielding a sieved number  $s$  of candidates;

performing an associated first one of said  $t$  primality test on each of said sieved number  $s$  of candidates, each of the plurality of  $s$  first primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $s$  of the exponentiation units, said first exponentiation operations being performed by said plurality of  $s$  exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number  $r$  of candidates; and

performing a plurality of  $t-1$  additional ones of said  $t$  primality tests on each of said remaining number  $r$  of candidates, each of the plurality of  $(r \times (t-1))$  first primality tests including an associated exponentiation operation executed by an associated one of a plurality of  $(r \times (t-1))$  of the exponentiation units, said  $(r \times (t-1))$  exponentiation operations being performed by said plurality of  $(r \times (t-1))$  exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

28. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching-in-parallel for a plurality of prime number values simultaneously, comprising the steps of:

randomly generating a plurality of  $k$  random odd numbers expressed as  $n_{0,0}, n_{1,0}, \dots, n_{(k-1),0}$ , each said number providing a prime number candidate;

determining a plurality of  $y$  additional odd numbers based on each one of the randomly generated odd numbers  $n_{1,0}, \dots, n_{(k-1),0}$  to provide  $(k \times y)$  additional prime number candidates